Claims

1. A system for improving the parallelization of image processing, using one or more parallelization modes, wherein said image that is displayed on at least one computer screen by one or more Graphic Processing Units (GPUs), comprising at least:

a. one or more software applications, for issuing graphics commands;

b. one or more graphic libraries, for storing data used to implement said graphics commands;

c. one or more Software Hub Drivers, for controlling a Hardware Hub, for interacting with the operation system of said computer and said graphic libraries, for performing real-time analysis of a data stream, from which frames of said image are generated, for determining the parallelization mode of each GPU, and for forwarding said data stream or a portion thereof to each GPU;

d. one or more GPU Drivers, for allowing said GPUs to interact with said graphic libraries; and

e. at least one I/O module for interconnecting between said Software module and said Hardware Hub,

wherein, said Hardware Hub distributes, for each frame, between said GPUs, graphic commands and said data stream or a portion thereof, and composites a graphics output for display, using the outputs obtained from at least one GPU, while reconfiguring, whenever required, said parallelization mode of each said GPU, for said each frame.

2. A system according to claim 1, wherein the parallelization is based on an object division mode or on an image division mode or on a time division mode or on any combination thereof.

3. A system according to claim 1, wherein the Hardware Hub comprises at least one compositing unit at least for composing a complete frame from processed portions of the data stream.

4. System according to claim 2, wherein the Hardware Hub comprises at least one hub router for routing polygonal data, for routing graphic command stream, for routing pixel data and for routing the results of composition, while operating in the object division mode or in the image division mode or in the time division mode or in any combination thereof.

5. System according to claim 1, wherein the Hardware Hub comprises at least one control unit for receiving commands from the Software Hub Driver within the I/O module.

6. System according to claim 1, wherein the Hardware Hub comprises a memory unit for storing intermediate processing results of one or more GPUs and data required for composition and transferring the processed data for display.

7. System according to claim 1, wherein the Software Hub Driver is capable of performing the following operations:

a. interception of the graphic commands from the standard graphic library by means of the OS and utilities;

b. forwarding and creating graphic commands to the GPU Driver by means of the OS interface and utilities;

c. controlling the Hardware Hub, registry and installation operations by means of the OS interface and utilities:

d. maintaining the consistency of graphic machine states across the GPUs, based on the input graphic commands stream, while using state monitoring;

e. estimating the type of graphic load and overload in the executed application graphic context, while using application and graphic resources analysis;

f. load estimation of the GPUs load balance based on graphic commands stream and time measurements, while using application and graphic resources analysis;

g. adjusting the load distribution between GPUs according to feedback received from each GPU regarding said load balance, while using application and graphic resources analysis;

h. performing manipulation in graphic functions according to the current parallelization mode; and

i. controlling the distributed graphic functions, while modifying said graphic commands and said data stream according to said current parallelization mode.

8. A method for improving the parallelization of image processing, using one or more parallelization modes, wherein said image that is displayed on at least one computer screen by one or more Graphic Processing Units (GPUs), comprising at least:

a. providing one or more software applications, for issuing graphics commands;

b. providing one or more graphic libraries, for storing data used to implement said graphics commands;

c. providing one or more Software Hub Drivers, for controlling a Hardware Hub, for interacting with the operation system of said computer and said graphic libraries, for

performing real-time analysis of a data stream, from which frames of said image are generated, for determining the parallelization mode of each GPU, and for forwarding said data stream or a portion thereof to each GPU; and

d. providing one or more GPU Drivers, for allowing said GPUs to interact with said graphic libraries.

e. providing at least one I/O module for interconnecting between said Software Hub Drivers and said Hardware Hub; and

f. for each frame, distributing between said GPUs and by means of said Hardware Hub, graphic commands and said data stream or a portion thereof, and compositing a graphics output for display, using the outputs obtained from at least one GPU, while reconfiguring, whenever required, said parallelization mode of each said GPU, for said each frame.

9. A method according to claim 8, wherein when the parallelization mode is an object division parallelization mode, ~~and~~ the following steps are performed:

a. for each frame, generating a stream of graphic operations and polygonal data;

b. marking ~~the~~ polygonal data and graphic commands by means of the Software Hub Driver for distribution between multiple GPUs;

c. sending the marked data to the Hardware Hub;

d. distributing said marked data via the Hardware Hub to said multiple GPUs;

e. rendering the data by means of GPUs;

f. retrieving the data from frame buffers associated with said GPUs and forwarding the retrieved data to the compositing unit via the Hardware Hub;

g. compositing the content of said frame buffers into a single frame buffer; and

h. forwarding the content of said single frame buffer to at least one designated GPU for display.

10. A method according to claim 8, wherein when the parallelization mode is an Image division parallelization mode, the following steps are performed:

a. subdividing the screen to portions and assigning different viewports to GPUs by means of the Software Hub Driver;

b. moving the entire polygonal data and graphic commands to the Hardware Hub;

c. transmitting said entire polygonal data and graphic commands to GPUs, wherein each GPU receives the same data;

d. rendering the data by means of GPUs;

e. forwarding a portion of the content stored in frame buffers associated with said GPUs to compositing unit in Hardware Hub for the complete image creation; and

f. forwarding said image to at least one designated GPU for display.

11. A method according to claim 8, wherein when the parallelization mode is a Time division parallelization mode, the following steps are performed:

a. forwarding to each one of the multiple GPUs the entire amount of polygons for rendering;

b. redirecting the entire polygonal data and graphic commands by means of Software Hub Driver to all GPUs, while alternating between them;

c. rendering the data by means of GPUs;

d. transferring rendered data from at least one GPU via the Hardware Hub; and

e. redirecting the resulting content of a frame buffer associated via the Hardware Hub to at least one designated GPU for display.

12. A method according to claim 8, wherein the distribution of polygons between multiple GPUs is performed by:

a. distributing blocks of data between multiple GPUs;

b. testing each graphic operation for blocking mode, in which one or more parallelization modes are carried out;

c. redirecting the data in regular non-blocking path to at least one designated GPU;

d. repeating step (b) and (c) until a blocking operation is detected;

e. synchronizing GPUs by the following sequence:

e.1. performing a flush operation in order to terminate rendering and clean up the internal pipeline in each GPU;

e.2. performing a composition operation for merging the contents of frame buffers associated with said GPUs into a single frame buffer; and

e.3. transmitting said single frame buffer back to all GPUs;

f. terminating the composited complete frame at all GPUs, except one or more designated GPUs, whenever a Swap operation is detected and displaying the image by means of said one or more designated GPUs;

g. processing the same data by means of all GPUs, as long as the blocking mode is active and the Swap operation is not detected; and

h. continuing to process the designated data by means of multiple GPUs, whenever the blocking mode is inactive.

13. A method according to claim 8, wherein the parallelization is based on an object division mode or on an image division mode or on a time division mode or on any combination thereof.

14. A system of claim 1, wherein said Hardware Hub distributes, for each frame, between said GPUs, graphic commands and said data stream or a portion thereof, according to their relative complexity within said image, said complexity is defined by said Software Hub Driver, and composites a graphics output for display, using the outputs obtained from at least one GPU, while reconfiguring, whenever required, said parallelization mode of each said GPU, for said each frame.

15. A method of claim 8, wherein said Hardware Hub distributes, for each frame, between said GPUs, graphic commands and said data stream or a portion thereof, according to their relative complexity within said image, said complexity is defined by said Software Hub Driver, and composites a graphics output for display, using the outputs obtained from at least one GPU, while reconfiguring, whenever required, said parallelization mode of each said GPU, for said each frame.

16. A computer system capable of displaying images of 3-D objects modeled within computer system, comprising:

(1) an operating system (OS);

(2) an I/O module;

(3) a graphics processing subsystem interfaced with said I/O module;

(4) one or more software applications for generating a stream of geometrical data and graphics commands supporting (i) the modeling of an object having 3D geometrical characteristics and (ii) the viewing of images of said object during an interactive process between said computer system and a user thereof;

(5) one or more graphic libraries for storing data used to implement said stream of geometrical data and graphics commands;

(6) at least one display surface for displaying said images by graphically displaying frames of pixel data produced by said graphics processing subsystem; and

said graphics processing subsystem including:

(a) a plurality of Graphic Processing Units (GPUs) arranged in a parallel architecture and operating according to one or more parallelization modes of operation so that each said GPU is allowed to process data in a parallel manner on the time, image and object domains, said one or more parallelization modes of operation including (i) a time division mode wherein each GPU renders a different frame of pixel data to be displayed at a different moment of time, (ii) an image division mode wherein each GPU renders a subset of the pixels used to compose each frame of pixel data to be displayed, and (iii) an object division mode wherein the object which is to be displayed as a frame of pixels, is decomposed into said stream of geometrical data and graphic commands which are distributed to said GPUs for rendering the frames of pixel data compositing the images to be displayed on said at least one display surface, wherein said frames of pixel data are (i) generated by said GPUs processing the stream of geometrical data and graphical commands while operating in one or more of said parallelization modes, and (ii) displayed on said at least one display surface by one or more of said GPUs;

(b) a Hardware Hub, interfacing with said I/O module and with said GPUs, for distributing the decomposed stream of geometrical data and graphic commands among said GPUs, compositing pixel data output from said GPUs for display according to different parallelization modes, and managing said parallelization mode of each said GPU in the parallel architecture;

(c) one or more software hub drivers, installed on said computer system, (1) for performing the following functions: (1) controlling said hardware hub, (2) interacting with the OS and said graphic libraries, (3) performing real-time analysis of said stream of geometrical data and said graphics commands, from which frames of said pixel data are generated, (4) determining the parallelization mode of each said GPU, and (5) forwarding said graphical commands and said geometrical data stream or a portion thereof to each said GPU; and

(d) one or more GPU drivers, installed on said computer system, for allowing said GPUs to interact with said graphic libraries;

wherein, for each image of said object to be generated and displayed, said hardware hub distributes between said GPUs, said graphic commands and said stream of geometrical data or a portion thereof, and composites a pixel data output using the pixel data outputs obtained from at least one said GPU, so as to generate a corresponding frame of pixel data to be displayed on said at least one display screen.

17. The computer system of claim 16, wherein said Hardware Hub manages said parallelization mode of each said GPU in the parallel architecture for the next frame of pixel data to be generated and displayed based on factors including the time required to render the previous

frames of pixel data and the bottlenecks exhibited in vertex processing and pixel processing during the rendering of said previous frames of pixel data.

18. The computer system of claim 16, wherein the parallelization mode operation is based on any combination of said object division mode, said image division mode and said time division mode.

19. The computer system of claim 18, wherein the Hardware Hub comprises at least one compositing unit at least for composing a complete frame of pixel data from processed portions of geometrical data stream.

20. The computer system of claim 16, wherein said geometrical data stream comprises polygonal data, and wherein the Hardware Hub comprises at least one hub router (i) for routing said polygonal data, (ii) for routing said graphic command stream, (iii) for routing pixel data and (iv) for routing the results of pixel composition, while said GPUs are operating in the object division mode, in the image division mode, or in the time division mode or in any combination thereof.

21. The computer system of claim 16, wherein the Hardware Hub comprises at least one control unit for receiving commands from the software hub drivers.

22. The computer system of claim 16, wherein the Hardware Hub comprises a memory unit for storing intermediate processing results of one or more GPUs and data required for composition and transferring the processed data for display.

23.The computer system of claim 17, wherein the software hub driver is capable of performing operations selected from the group consisting of:

(1) interception of the graphic commands from the standard graphic library by means of the OS and utilities;

(2) forwarding and creating graphic commands to the GPU Driver by means of the OS and utilities; and

(3) controlling the Hardware Hub, registry and installation operations by means of the OS interface and utilities:

(4) maintaining the consistency of graphic machine states across the GPUs, based on the input graphic commands stream, while using state monitoring;

(5) estimating the type of graphic load and overload in the executed application graphic context, while using application and graphic resources analysis;

(6) load estimation of the GPUs load balance based on the graphic data and commands stream and time measurements, while using application and graphic resources analysis;

(7) adjusting the load distribution between GPUs according to feedback received from each GPU regarding said load balance, while using application and graphic resources analysis;

(8) performing manipulation of graphic functions performed by said GPUs, according to the current parallelization mode; and

(9) controlling the distributed graphic functions performed by said GPUs, while modifying said graphic commands and said geometrical data stream according to said current parallelization mode.

24. The computer system of claim 16, wherein said object is decomposable into a plurality of polygons, and wherein said geometrical data comprises the vertices of said polygons.

25. The computer system of claim 16, wherein each said GPU comprise a corresponding frame buffer for storing a frame of pixel data generated by said GPU.

26. The computer system of claim 21, wherein each pixel associated with a frame of pixel data includes attributes selected from the group consisting of color, alpha, position, depth, and stencil.

27. A method of generating frames of pixel data of an object for display using a computer system including (1) an operating system (OS), (2) an I/O module, (3) one or more software applications for generating a stream of geometrical data and graphics commands supporting (i) the modeling of an object having 3D geometrical characteristics and (ii) the viewing of images of said object during an interactive process between said computer system and a user thereof, (4) one or more graphic libraries for storing data used to implement said stream of geometrical data and graphics commands, and (5) at least one display surface for displaying said images by graphically displaying frames of pixel data produced by said graphics processing subsystem, said method comprising the steps of:

(a) interfacing a Hardware Hub with said I/O module;

(b) interfacing with said hardware hub, a plurality of Graphic Processing Units (GPUs) arranged in a parallel architecture and operating according to one or more parallelization

modes of operation so that each said GPU is allowed to process data in a parallel manner on the time, image and object domains, said one or more parallelization modes of operation including (i) a time division mode wherein each GPU renders a different frame of pixel data to be displayed at a different moment of time, (ii) an image division mode wherein each GPU renders a subset of the pixels used to compose each frame of pixel data to be displayed, and (iii) an object division mode wherein the object which is to be displayed as a frame of pixel, is decomposed into said stream of geometrical data and graphic commands which are thereafter distributed to said GPUs for rendering the frames of pixel data compositing the images to be displayed on said at least one display surface, and each said GPU having a corresponding frame buffer for storing a frame of pixel data generated by said GPU;

(c) installing within said computer system, one or more software hub drivers, for performing the following functions: (1) controlling said Hardware Hub, (2) interacting with said OS and said graphic libraries, (3) performing real-time analysis of said graphic commands and said geometrical data stream, from which frames of pixel data are generated, (4) determining the parallelization mode of each GPU, and (5) forwarding said graphic commands and said geometrical data stream or a portion thereof to each GPU;

(d) installing within said computer system, one or more GPU Drivers, for allowing said GPUs to interact with said graphic libraries;

(e) for each image of said object to be generated and displayed, said Hardware Hub distributing between said GPUs, said graphic commands and said stream of geometrical data or a portion thereof, said GPUs processing the stream of geometrical data and graphical commands while operating in said one or more of said parallelization modes, and compositing a pixel data output using the pixel data outputs obtained from at least one said GPU, so as to generate a corresponding frame of pixel data to be displayed on said at least one display screen.

28. The method of claim 27, which further comprises:

(f) said Hardware Hub managing said parallelization mode of each said GPU for the next frame of pixel data to be generated and displayed, said management of aid parallelization mode of each said GPU being based on factors including the time required to render the previous frames of pixel data and the bottlenecks exhibited during the rendering of said previous frames of pixel data.

29. The method according to claim 28, wherein when the parallelization mode is the object division mode, the following steps are performed:

(1) for each frame of pixel data to be composited, generating a stream of graphic operations and polygonal data;

(2) marking the polygonal data and graphic commands by means of the software hub driver, for distribution between said GPUs;

(3) sending the marked polygonal and graphic command data to said Hardware Hub;

(4) distributing said marked polygonal and graphics command data via the hardware hub to said GPUs;

(5) rendering the data by means of said GPUs;

(6) retrieving the data from the frame buffers and forwarding the retrieved data to the compositing unit via the Hardware Hub;

(7) compositing the content of said frame buffers into a single frame buffer; and

(8) forwarding the content of said single frame buffer to at least one designated GPU for display.

30. The method of claim 28, wherein when the parallelization mode is an image division parallelization mode, the following steps are performed:

(1) subdividing the screen to portions and assigning different viewports to GPUs by means of the software hub driver;

(2) moving the entire polygonal data and graphic commands to the Hardware Hub;

(3) transmitting said entire polygonal data and graphic commands to GPUs, wherein each GPU receives the same data;

(4) rendering the data by means of GPUs;

(5) forwarding a portion of the content stored in the frame buffers to compositing unit in said hardware hub for the complete image creation; and

(6) forwarding said image to at least one designated GPU for display.

31. The method of claim 28, wherein when the parallelization mode is a time division parallelization mode, the following steps are performed:

(1) forwarding to each one of the multiple GPUs the entire amount of polygons for rendering;

(2) redirecting the entire polygonal data and graphic commands by means of software hub driver to all GPUs, while alternating between them;

(3) rendering the data by means of GPUs;

(4) transferring rendered data from at least one GPU via the Hardware Hub; and

(5) redirecting the resulting content of the frame buffer via Hardware Hub to at least one designated GPU for display.

32. The method of claim 28, wherein the distribution of polygons between multiple GPUs is performed by:

(1) distributing blocks of data between multiple GPUs;

(2) testing each graphic operation for blocking mode, in which one or more parallelization modes are carried out;

(3) redirecting the data in regular non-blocking path to at least one designated GPU;

(4) repeating step (b) and (c) until a blocking operation is detected;

(5) synchronizing GPUs by the following sequence:

(5a) performing a flush operation in order to terminate rendering and clean up the internal pipeline in each GPU;

(5b) performing a composition operation for merging the contents of the frame buffers into a single frame buffer; and

(5C) transmitting said single frame buffer back to all GPUs;

(6) terminating the composited complete frame at all GPUs, except one or more designated GPUs, whenever a Swap operation is detected and displaying the image by means of said one or more designated GPUs;

(7) processing the same data by means of all GPUs, as long as the blocking mode is active and the Swap operation is not detected; and

(8) continuing to process the designated data by means of said GPUs, whenever the blocking mode is inactive.

33. A graphics processing subsystem for use in a computer system capable of displaying images of 3-D objects modeled therewithin, said computer system including

(1) an operating system (OS);

(2) an I/O module;

(3) one or more software applications for generating a stream of geometrical data and graphics commands supporting (i) the modeling of an object having 3D geometrical characteristics and (ii) the viewing of images of said object during an interactive process between said computer system and a user thereof;

(4) one or more graphic libraries for storing data used to implement said stream of geometrical data and graphics commands;

(5) at least one display surface for displaying said images by graphically displaying frames of pixel data produced by said graphics processing subsystem, and said graphics processing subsystem comprising:

(a) a plurality of Graphic Processing Units (GPUs) arranged in a parallel architecture and operating according to one or more parallelization modes of operation so that each said GPU is allowed to process data in a parallel manner on the time, image and object domains, wherein frames of pixel data for display on said display surface are (i) generated by said GPUs processing the stream of geometrical data and graphical commands while operating in one or more of said parallelization modes, and (ii) displayed on said at least one display surface by one or more of said GPUs;

(b) a Hardware Hub, interfacing with said I/O module and with said GPUs, for distributing the decomposed stream of geometrical data and graphic commands among said GPUs, compositing pixel data output from said GPUs for display according to different parallelization modes, and managing said parallelization mode of each said GPU in the parallel architecture;

(c) one or more software hub drivers, (1) for performing the following functions: (1) controlling said Hardware Hub, (2) interacting with the OS and said graphic libraries, (3) performing real-time analysis of said stream of geometrical data and said graphics commands, from which frames of said pixel data are generated, (4) determining the parallelization mode of each said GPU, and (5) forwarding said graphical commands and said geometrical data stream or a portion thereof to each said GPU; and

(d) one or more GPU drivers, installed on said computer system, for allowing said GPUs to interact with said graphic libraries; and

wherein, for each image of said object to be generated and displayed, said Hardware Hub distributes between said GPUs, said graphic commands and said stream of geometrical data or a portion thereof, and composites a pixel data output using the pixel data outputs obtained from at least one said GPU, so as to generate a corresponding frame of pixel data to be displayed on said at least one display screen.

34. The graphics processing subsystem of claim 33, wherein said Hardware Hub manages said parallelization mode of each said GPU in the parallel architecture for the next frame of pixel data to be generated and displayed based on factors including the time required to render the previous frames of pixel data and the bottlenecks exhibited during the rendering of said previous frames of pixel data.

35. The graphics processing subsystem of claim 34, wherein said one or more parallelization modes of operation comprise: (i) a time division mode wherein each GPU renders a different frame of pixel data to be displayed at a different moment of time, (ii) an image division mode wherein each GPU renders a subset of the pixels used to compose each frame of pixel data to be displayed, and (iii) an object division mode wherein the object which is to be displayed as a frame of pixels, is decomposed into said stream of geometrical data and graphic commands which are distributed to said GPUs for rendering the frames of pixel data compositing the images to be displayed on said at least one display surface.

36. The graphics processing subsystem according to claim 33, wherein said software hub drivers are installed on said computer system.

37. The graphics processing subsystem according to claim 33, wherein said one or more GPU drivers are installed on said computer system.

38. The graphics processing subsystem of claim 33, wherein said object is decomposable into a plurality of polygons, and wherein said geometrical data comprises the vertices of said polygons.